Visible Light Localization Using Conventional Light Fixtures and Smartphones

Chi Zhang, Xinyu Zhang, *Member, IEEE*

Abstract—Owing to dense deployment of light fixtures and multipath-free propagation, visible light localization technology holds potential to overcome the reliability issue of radio localization. However, existing visible light localization systems require customized light hardware, which increases deployment cost and hinders near-term adoption. In this paper, we propose LiTell, a simple and robust localization scheme that employs unmodified fluorescent lights (FLs) as location landmarks and commodity smartphones as light sensors. LiTell builds on the key observation that each FL has an inherent characteristic frequency which can serve as a discriminative feature. It incorporates a set of sampling, signal amplification, and camera optimization mechanisms, that enable a smartphone to capture the extremely weak and high-frequency (greater than 80 kHz) features. We have implemented LiTell as a real-time localization and navigation system on Android. Our experiments demonstrate LiTell's high reliability in discriminating different FLs, and its potential to achieve sub-meter location granularity. Our user study in a multi-storey office building, parking lot, and grocery store further validates LiTell as an accurate, robust and ready-to-use indoor localization system.

Index Terms—Visible light sensing, Visible light localization, Indoor localization, Smartphones, Signal processing, Image processing.

1 Introduction

Indoor localization technology is bringing huge impacts on human activities, in the same way that GPS did in revolutionizing outdoor navigation. Existing market research predicts that location-based services in retail industry alone will generate 10 billion revenues by 2020 [1]. However, after decades of research, there still lacks a solution with desired *simplicity* and *robustness*. Recent field tests of state-of-the-art localization schemes [2], [3] revealed a common set of problems including high deployment overhead and low reliability, concluding that robust meter-level indoor localization remains an open problem, even in a small sandbox environment (300 m^2) with simple layout [2].

Radio-based localization techniques are most extensively studied due to readily-available infrastructure. Prior research has explored fingerprinting [4], [5], [6], propagation modeling [7], and triangulation [8], [9]. However, the elusive nature of radio signals renders them less reliable in real environments, specifically due to three major challenges. *(i)* Multipath reflections. Multipath reflections are strongly dependent on the geometries and construction materials in indoor environments, which defeats model-based approaches. *(ii)* Environment dynamics. Minor change in the environment (*e.g.*, adding a new furniture, human presence and mobility) can substantially disturb the received signals strength (RSS) and phase, thus compromising location estimation [2]. Even the body orientation and holding position of devices (which block the antenna in different ways) can cause 6 to 8 meters of location error [10]. *(iii)* Device heterogeneity. Depending on model, co-located smartphone devices can observe up to 10 dB RSS differences [11]. In enterprise environments, access points' dynamic channel selection and power control further vary the signal features.

Visible light (VL) localization techniques hold potential to overcome such fundamental limitations. Using ceiling-mounted LEDs as beaconing devices, VL localization can achieve sub-meter precision [12], [13], and can even determine the orientation of a smartphone [14]. However, almost all existing LED-based localization techniques require customized beaconing circuits to be added to the LED driver [15], which involve substantial retrofitting cost and are unlikely to be adopted pervasively in the near term [16]. IDyLL [17] reuses existing lights as periodic landmarks to complement dead-reckoning. However, the accuracy of dead-reckoning becomes the performance bottleneck and the error often rises to 8–10 m. Alternatively, low-cost barcodes can be used for localization [18], but they are visually obtrusive and requires additional deployment effort as well.

Inspired by research in electromagnetic interference (EMI) detection [19], [20], we propose *LiTell*, a simple and robust VL localization scheme that readily works with unmodified light fixtures and commercial-off-the-shelf (COTS) smartphones. LiTell uses incumbent fluorescent lights (FLs) as location anchors, and smartphones as receivers. The key hypothesis is that an FL's driver acts as an oscillator with a resonance frequency. Due to unavoidable manufacturing variation, different FLs have different resonance frequencies, which will cause each of them flickering at a characteristic frequency (CF), a high frequency ($>$ 80 kHz) that is imperceptible by human and remains relatively stable in practice. LiTell uses the CF as a discriminative feature among different FLs, which in turn serve as location landmarks.

We have tested the hypothesis using a customized high-speed light sensor. Our experiments demonstrate that the CFs are highly *diverse*: among 500 FLs in an office building, over 99% have a pairwise CF separation of more than

- *C. Zhang is with the Department of Electrical and Computer Engineering, University of California San Diego, CA, 92093.*
E-mail: c4zhang@ucsd.edu
- *X. Zhang is with the Department of Electrical and Computer Engineering, University of California San Diego, CA, 92093.*
E-mail: xyzhang@ucsd.edu

Manuscript received XXXX XX, XXXX; revised XXXX XX, XXXX.

20 Hz. The CF is also highly *reliable*: it is unaffected by typical environment dynamics or human behavior, and varies by only a few tens of Hz across several months. To our knowledge, this represents the first study to characterize the FLs' optical frequency feature and use it to discriminate FLs.

However, many challenges emerge when sampling CF with COTS smartphone cameras: *(i) Low sampling rate*. cameras are designed for snapshotting low-motion scenes at around 30 frames per second (FPS). The rolling shutter mechanism, which allows a camera to capture several thousand samples within one frame [14], [21], may alleviate the problem, but the sampling rate remains insufficient. *(ii) Low sensitivity to high-frequency light signals*. Camera sensors have a limited dynamic range, majority of which is already occupied by low-frequency but high-power (bright) signals when capturing a real scene. Besides, substantial noise exists in high frequencies, introduced by camera hardware (*e.g.*, heterogeneity of color pixels, salt-and-pepper noise) and spatial patterns of physical structures (*e.g.*, latticed covers around FLs). These factors immerse the CF signals in the camera image's noise floor.

To overcome camera's low sampling rate, LiTell's solution builds on two observations. First, the CF is extremely sparse, representing a single peak in the frequency spectrum. Second, although cameras snapshot low-rate scenes, their analog bandwidth can reach a few hundred kHz. Thus, when sampled at a low rate, the CF will be "folded" back to low frequencies due to aliasing [22]. By optimizing the camera's sampling mechanics (*e.g.*, exposure time), along with the rolling shutter effect, we are able to recover CFs well above 80 kHz. In addition, to combat the low-sensitivity issue, we design a set of feature amplification mechanisms that synthesize consecutive captures of the FL, isolate the interference from ambient spatial patterns and mitigate the noise from camera hardware. These mechanisms substantially boost the SNR, allowing the CF to be easily identified in the spectrum.

LiTell's location matching mechanism is fairly simple. We first use a light sensor to measure the ground-truth CF of each FL luminary, and store the ⟨location, CF⟩ pair in a light registration database. The registration overhead is bounded by *number of light fixtures*, each taking only a few seconds, instead of all possible locations. At run time, any smartphone can capture images of a nearby FL, run the above sampling and amplification mechanisms to identify the CF, and look up the location from the database. To counteract occasional CF feature collisions and drifts, LiTell uses every two consecutive FLs as a location landmark. To ensure scalability, LiTell can optionally fuse with other landmarks such as WiFi, further limiting potential collisions. These afford *light-cell level localization*, which translates into meter-level granularity, considering the pervasive and dense deployment of FLs in typical public buildings. In addition, LiTell takes advantage of the scaling relation between the physical size and image size of an FL, and use simple geometrical model to derive the smartphone's position relative to the FL. This enables finer-grained localization with *sub-meter precision*.

We have implemented LiTell as a mobile app on Android smartphones, and also prototyped an indoor navigation app atop. Our microbenchmark experiments in a 4-storey office building demonstrate that LiTell can identify the CF features with high reliability. The identified features stay within 20 Hz of the ground-truth, with up to 2 m light-to-phone distance, under various usage behaviors and environment conditions, and across multiple generations of Android phone models built from 2012 to 2015, on both front and back cameras.

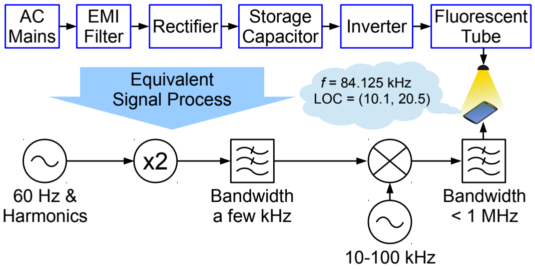


Fig. 1. General circuitry model of FL.

To verify LiTell's localization accuracy, reliability and usefulness in the wild, we conduct a user study with 10 volunteer participants. The user study took place in 3 uncontrolled environments: an office building (∼9000 m^2, 119 FLs), an indoor parking lot (∼2800 m^2, 91 FLs), and a grocery store (∼1000 m^2, 162 FLs). LiTell demonstrates the following key features: *(i)* Accuracy and usability. LiTell identifies the FLs with an average accuracy of 90.3% across the 3 sites, which can reach over 98% with the help of WiFi. Typically, it navigates the user to randomly selected POIs with 50% shorter time and 60% shorter distance consistently, compared with using visual landmarks and labels. It adds marginal overhead, compared with an oracle walking trial (directly towards destination with known routes). *(ii)* Robustness. By using the CF as signature, LiTell's localization accuracy becomes unaffected by heavy human activities and environment dynamics. To our knowledge, LiTell represents the first ready-to-use, real-time localization system to achieve a combination of such desirable traits.

The main contributions of LiTell can be summarized as follows[1]: *(i)* We conduct the first comprehensive feasibility study to verify the optical CF as a *diverse* and *reliable* feature to discriminate FLs. *(ii)* We design a set of sampling and signal amplification mechanisms that allow a COTS smartphone to capture the CF feature and single it out from various noises. *(iii)* We develop simple schemes that enable LiTell to achieve robust and accurate localization, with light level and sub-light level location granularity. The real-time version of LiTell system is implemented on Android, and verified in uncontrolled, multi-floor environment.

2 Characterizing Fluorescent Lights

2.1 Fluorescent Light and Its Frequency Components

A fluorescent light (FL) produces visible light by striking an arc across a tube lamp, causing the gas and fluorescent material inside to glow. The amount of current passing through the lamp determines the light intensity. FL driver acts as the core circuitry for light generation. It converts the AC mains voltage (110/220 V, 50/60 Hz) to a high-frequency AC voltage to sparkle the fluorescent tubes [24].

Fig. 1 depicts a general model of FL driver. The AC mains voltage first goes through some electromagnetic interference (EMI) filters. The rectifier then converts it to DC plus a residual AC component, whose frequency is doubled

[1] A preliminary version of the work appeared in [23].

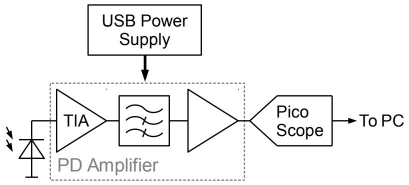


Fig. 2. Experimental setup to measure FLs' frequency features.

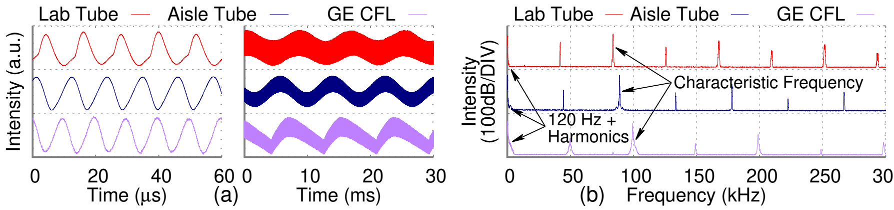


Fig. 3. Light intensity (a) waveforms at different time scales and (b) frequency components of FL.

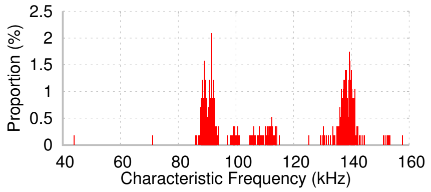


Fig. 4. Distributions of characteristic frequencies of FLs in the office building.

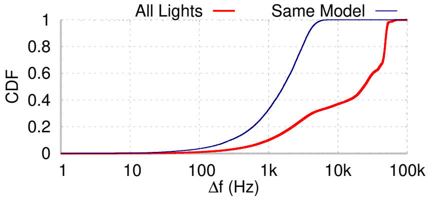


Fig. 5. CDF of differences in frequencies (Δf) between all FLs and FLs of the same model.

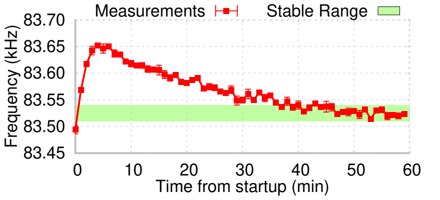


Fig. 6. Characteristic frequency converging during the start-up process of one tube FL.

from the 50/60 Hz AC mains frequency to 100/120 Hz. Afterwards, the core of the FL driver, an *inverter*, modulates this DC into high-frequency AC to drive the fluorescent tube. The resonance frequency of the inverter is determined by a group of components [25]. Due to manufacturing variation, values of these components usually vary within 5–20% range [26]. As a result, even among FLs of the same model, their resonance frequencies tend to vary significantly. The fluorescent tube then converts the high-frequency AC current (generated by the inverter) into visible light. Due to the frequency response of fluorescence [27], harmonics more than a few MHz are nearly completely filtered out. In general, the observable frequencies usually span a wide range from above 40 kHz to 1 MHz.

Note that the components in the inverter also have large temperature coefficients. This can cause the frequency to drift by over 2000 ppm across the typical temperature range (*e.g.* 0–70 °C) [28]. However, since the temperature in most public buildings is regulated, the temperature of the inverter should also be stable, leading to actual frequency stability several folds better. In what follows, we will verify the stability through comprehensive measurements.

2.2 Feasibility of FL Identification

Based on the operating principle of FL circuitry, *we hypothesize that individual FLs can be discriminated by their frequency characteristics*. This section verifies this hypothesis through comprehensive experiments, focusing on the uniqueness and stability of frequency features. Fig. 2 illustrates our measurement setup. We sample the FLs' optical emissions using a customized high-speed light sensor comprising a `BPW34` photodiode (PD) and a 2-stage amplifier, whose analog bandwidth is 1.6 MHz and output digitized by a portable oscilloscope (PicoScope 2207A [29]).

2.2.1 Uniqueness of Frequency Characteristics

Frequency characteristics of FL's optical emission. Fig. 3(a) plots the time series of light intensity from several example FLs, including tube lights and compact FLs (CFLs) in an office building. Observe that all the FLs' waveforms manifest quasi-periodic patterns, both at μs and ms scales. The periodicity becomes obvious in frequency domain (Fig. 3(b)). More specifically, all the FLs have a fundamental frequency component within the 40–60 kHz range, followed by its harmonics (*i.e.*, integer multiples). Notably, the fundamental frequency of each light can differ by a few kHz, which implies the feasibility of discrimination. The 120 Hz AC component (doubled from 60 Hz by rectification) and its harmonics are also visible in the spectrum, but only span a few kHz in total.

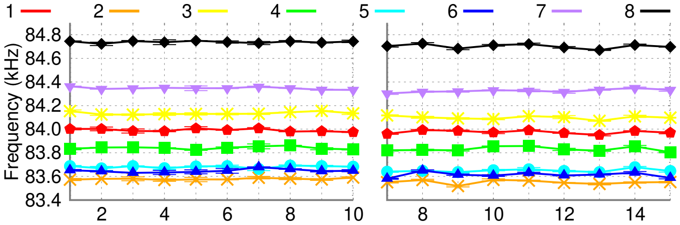
It is worth noting that the dominant frequency always occurs at 2× the fundamental frequency, and has a 20–30 dB higher magnitude. This is because the inverter creates different gains at the positive and negative halves of its sine waves, resulting in a weaker frequency component with half of the frequency [27]. As we will show in Sec. 4, when using a smartphone, it is usually impossible to see the features except for the strongest. Thus, *we select the dominant frequency as the* ***characteristic frequency (CF)*** *of the FL.*

Diversity of CF. To see how diverse the CFs can be, we measured over 500 tube FLs and CFLs in our 4-floor office building. Fig. 4 plots the histogram of the CFs. The results imply multiple types of FLs, with most CF features ranging from 80–160 kHz. Each type's CFs approximate a normal distribution. Fig. 5 further plots the CDF of pairwise differences in CF (denoted as Δf) among the FLs. We observe that only less than 0.1% of the pairs have $\Delta f \leqslant 10$ Hz, and 0.2% have $\Delta f \leqslant 20$ Hz, even among FLs of the same model. This confirms the CF as a strong feature to discriminate FLs at a large scale. In Sec. 5 and 7, we will provide analytical justifications for the probability of feature collision and implications for location discrimination.

2.2.2 Stability of Characteristic Frequency

Start-up profile of CF. Since the FL would heat up during startup, its temperature and CF should drift during the process. To characterize such drift, we cold start a tube FL and measure its CF across 1 hour. Fig. 6 plots the CF (error bars indicate std. across 16 snapshots at each 1-minute timestamp). The frequency first overshoots by a few hundred Hz, and then gradually decreases as the FL warms up. *However, after around 40 minutes, it converges to a stable CF with less than 20 Hz fluctuation.* For most commercial place and office buildings, the internal FLs tend to keep on for a long time after the start-up and remain heated-up, so the impact on the usability of CF as a discriminative feature should be negligible.

Temporal stability of characteristic frequency. To verify the temporal stability of CF, we measure the CF of 8



(a) Day (b) Week

Fig. 7. Short-term and long-term stability of characteristic frequency.

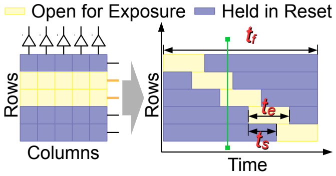


Fig. 8. A brief illustration of rolling shutter.

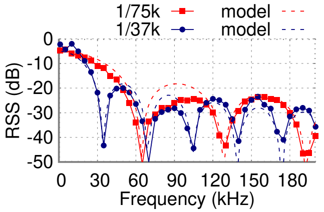


Fig. 9. Frequency response of a camera under various exposure time settings.

randomly selected FLs in our office building over short/long term. All 8 FLs are of the same model. Fig. 7(a) plots the CF values across 10 days, with 10 measurements per day per FL (error bars denote 90th-percentile values). The CFs demonstrate high stability, with variations within 20 Hz. Since the measurements span different time of the day, with different loads on the powerline, the results also imply that the CF features are robust against different powerline loads in large buildings.

We also evaluated the temporal stability over 15 weeks, sampling on a random day in each week. The results in Fig. 7(b) show slightly higher variation compared with the 10-day short-term variation, which also increases the risk of CF collision among FLs. As a result, *for robust localization using CF features, LiTell must be able to tolerate small amount of collisions.* Sec. 5 will introduce LiTell's error tolerance mechanisms.

3 Capturing CF Using Smartphones

Amid the photodiode sensors' potential in discriminating FLs' characteristic frequencies, fundamental challenges emerge when we try to capture such high-frequency features using smartphones, which are designed to capture static or low-rate scenes[2]. In this section, we describe how LiTell overcomes such challenges by exploiting camera as a generic optical channel sampler.

3.1 Sampling Light Using Smartphone Cameras

Due to considerations in cost and speed, modern CMOS image sensors used in smartphones usually expose different rows in the image at different time, while pixels within each row are exposed simultaneously (which can be summed into a single *sample*). This is called the rolling shutter effect [30] and can act as a sampling process with much higher sampling rate than the framerate. As a result, smartphones hold the potential for being used to sample high-frequency light signals. On the other hand, this sampling frequency is backed by high-quality hardware clock source, as opposed to software running on top of an operating system (OS) that controls the framerate. As a result, it is far more reliable and stable than the framerate. We have tested 2 Nexus 5 phones (with method in Sec. 4.1) and found their sampling rate to be extremely stable and consistent, varying only a few ppm (parts per million, or less than 1 Hz error for 75 kHz sampling rate) in typical conditions. This makes the rolling shutter a reliable sampling mechanism.

Fig. 8 shows a simplified diagram of CMOS image sensor and its sampling process. The image sensor's rows are opened for exposure sequentially. The delay between the opening of adjacent rows is called *sampling interval*, denoted as t_s. Duration of opening for each row is called *exposure time*, denoted as t_e. The time between the start of exposure of the first row and the end of the last row is the minimum frame time or *minimum frame interval*, denoted as t_f. Denote N as the number of rows, we have $t_s = t_f/N$. Inverting both sides, we can obtain the effective sampling rate of the rolling-shutter camera, which is $f_s = NR_f$.

Contemporary smartphones commonly support full HD capturing (1920×1080 resolution at $R_f = 30$ FPS frame rate). The corresponding lower bound of sampling rate is thus $f_s = 1080 \times 30 = 32.4$ Ksps, which is a thousand times higher than framerate, but still below the Nyquist sampling rate for typical FLs' characteristic frequencies ($> 2 \times 80$ KHz, Sec. 2). In the following sections, we introduce how LiTell recover CF with such insufficient sampling rate.

3.2 Preserving High-Frequency FL Features

The camera's sampling duration t_s depends on hardware and is unaffected by the exposure time setting t_e. In fact, $t_e \geqslant t_s$ since exposure of different row can overlap as shown in Fig. 8. However, the exposure process can be considered as an integration (or moving-average filter), so t_e does affect the analog channel response of the camera and determines its capability to capture high-frequency signals.

More specifically, at sampling time t (after the beginning of an exposure), the camera's output is:

$$A(t) = \int_{-\infty}^{+\infty} I(\tau) g(\tau - t) \mathrm{d}\tau \tag{1}$$

where $I(t)$ is the light intensity at time t, $g(\tau - t)$ is a gate (rectangular) function that evaluates to 1 during $(t, t + t_e)$ and 0 otherwise. Applying Fourier transform on both sides, we have:

$$\begin{aligned} \mathcal{F}(A) &= \int_{-\infty}^{+\infty} I(\tau) e^{-j2\pi f \tau} \mathrm{d}\tau \int_{-\infty}^{+\infty} g(-T) e^{-j2\pi f T} \mathrm{d}T \\ &= -\mathcal{F}(I) \cdot \mathcal{F}(g) \end{aligned} \tag{2}$$

where $T = t - \tau$. Now we can obtain the frequency response of the sampling process as:

$$|H(f)| = \frac{|\mathcal{F}(A)|}{|\mathcal{F}(I)|} = |\mathcal{F}(g)| = |\operatorname{sinc}(f t_e)| \tag{3}$$

In practice, photoelectrons continue to accumulate during the readout phase after exposure is completed [21]. Accordingly, the integration time should be $t_e + t_r$, where t_r is the readout duration. So Eq. (3) should be updated as:

$$|H(f, t_e)| = |\operatorname{sinc}[f(t_e + t_r)]| \tag{4}$$

To verify this frequency response model against actual smartphones, we use the arbitrary waveform generator on PicoScope to generate sine tones across a wide range of frequen-

[2] Note that smartphones' light sensors cannot capture the CF feature because their sampling frequencies are limited to a few Hz.

cies, which are then used to drive a white LED, serving as an ideal optical source. We capture the signals using a Nexus 5 phone, which has a sampling rate $f_s = 1/t_s = 75.1624$ Ksps according to our measurement (to be discussed in Sec. 3.3). Fig. 9 plots the RSS of the captured sine signals (relative to the DC baseline) across different frequencies, which reflects the frequency response of the camera. The results reveal that under a given exposure time setting, the camera's response follows the sinc function, which matches the above model. Whereas Nexus 5's advertised minimum exposure time $t_e = 1/75586$ s, the first notch of the sinc function appears at $\sim 1/65200$ s, *i.e.*, $t_e + t_r \approx 1/65200 \text{ s} > t_e$, which matches the refined model in Eq. (4).

Note that under a given exposure time setting, the camera's frequency response has notch points at $k/(t_e + t_r), k = 1, 2, \cdots$. If the characteristic frequency of an FL lies near a notch, then the frequency feature will become extremely weak. The *adaptive exposure* mechanism in LiTell is designed to overcome such situations. Specifically, LiTell selects the optimal exposure time t_e^* to maximize the camera's mean response to the characteristic frequencies of all FLs (*e.g.*, within the building of interest), *i.e.*,

$$t_e^* = \arg\max_i \sum_{j=1}^{L} |H(f_j, t_e^i)| \tag{5}$$

where L is the number of FLs. t_e^i indexes the exposure time of the i-th exposure setting. Since all notches can be avoided within limited number of exposure time settings, the optimization can be solved by LiTell's backend server offline through an exhaustive search. Note that t_r is a constant for each camera and can be factory calibrated.

3.3 Recovering High-Frequency FL Features

The foregoing section reveals that the frequency response of the camera remains high beyond the sampling rate. In other words, signals with frequency higher than $f_s/2$ can still be sampled, since the analog bandwidth of the camera is much higher than the digital sampling rate. LiTell leverages this property to recover high-frequency signals.

It is well known that when sampling a high-frequency signal at sub-Nyquist rate, the frequency component will be aliased or folded back [22]. The rule of aliasing is simple:

$$f_a = \begin{cases} (N+1)f_s - f & f_s/2 < f - Nf_s < f_s \\ f - Nf_s & 0 \leqslant f - Nf_s \leqslant f_s/2 \end{cases} \tag{6}$$

where f_a is the aliased frequency, f is the original frequency, f_s is the sampling frequency and $N = 0, 1, 2, \cdots$. Given f_s, we can derive a few candidates of f from the measured f_a:

$$f \in \{f_g | f_g = Nf_s \pm f_a,\ N = 0, 1, 2, \cdots,\ f_g \geqslant 0\} \tag{7}$$

where f_g is a candidate estimation of f. For example, $f_s = 75$ kHz and $f_a = 15$ kHz leads to $f \in \{15, 60, 90, 135, \cdots \text{ kHz}\}$, and for $f_a = 3$ we have $f \in \{3, 72, 78, 147, \cdots \text{ kHz}\}$. In practice we find all FLs have f well above 40 kHz and below 200 kHz, so frequencies beyond the range can be excluded.

Whereas the error in f_a will be carried to the estimation of f as is, the error in f_s can get amplified in the process when N is large. Since the difference among FLs' CF features can sometimes be as small as a few tens of Hz (Sec. 2), this calls for a precise calibration of f_s. In practice, this can be done in factory since the sampling frequency of the camera is derived from crystal oscillators and remains extremely stable over time.

In our experiments and implementation, we use PicoScope and an LED to send a 10 kHz calibration signal, keeping other settings identical as in Sec. 3.2. We then run MUSIC (an eigen spectrum analysis algorithm [31]) over the camera-captured signals to get a super-resolution estimation of the received normalized signal frequency f_n ($f_n < 0.5$). The camera's sampling frequency is then calibrated as: $f_s = 10 \text{ kHz}/f_n$.

4 Amplifying High-Frequency Features

Although it is feasible to sample the FL's high-frequency characteristic signals using a camera, the signals can be extremely weak — inherently, the FL is designed for zero-frequency illumination signals, and camera for snapshotting a still scene. In this section, we present a set of signal processing and camera optimization mechanisms in LiTell that together overcome these challenges.

4.1 Spatial Noise and Interference Suppression

Owing to the rolling shutter effect, LiTell can sample the same FL at high frequencies across different rows of an image. This assumes the FL is an ideal homogeneous light source and fully occupies the camera's FoV. But two practical challenges can break this premise: *(i)* "Salt-and-pepper" noise in dark area, though barely visible to human eyes, may create substantial noise at high frequencies. *(ii)* Strong ambient light (*e.g.*, sunlight reflected to ceiling) may also raise the noise floor. In LiTell, we address such challenges by isolating irrelevant pixels from the camera image, which we refer to as spatial noise and interference suppression (*SNIS*). We now describe SNIS' major steps.

Cleaning up image for reliable contour detection. Unlike modulated LEDs in previous works [14], [21], LiTell faces two new challenges: *(i)* it must be able to isolate closely placed FLs, which is common in many buildings; *(ii)* it must be able to process million-pixel images on smartphones with minimum latency. These challenges make it difficult to follow the exact procedures in [14], which applies blur to mitigate noise, and then threshold the image into a black-and-white mask to identify the light. We thus threshold [32] the image into a mask first, and then use morphological opening and closing [33], [34] on the mask, a classical image processing algorithm that recovers a single shape by connecting densely distributed dots, while removing noisy outliers. After such step, the mask will only contain one or a few large connected components, which speeds up light contour detection.

Generating a rigid contour. The morphological processing yields a single shape which can be isolated by finding the largest contour, but it leaves one problem unattended: the shape is irregular, causing different number of pixels in each row to be summed into the sample. Consequently, the gain provided by summing varies across samples, which introduces high-frequency artifacts in the resulting spectrum. To solve the problem, we create a rigid bounding box around the FL image following [34]. Since most light fixtures appear rectangular on cameras, this method effectively minimizes

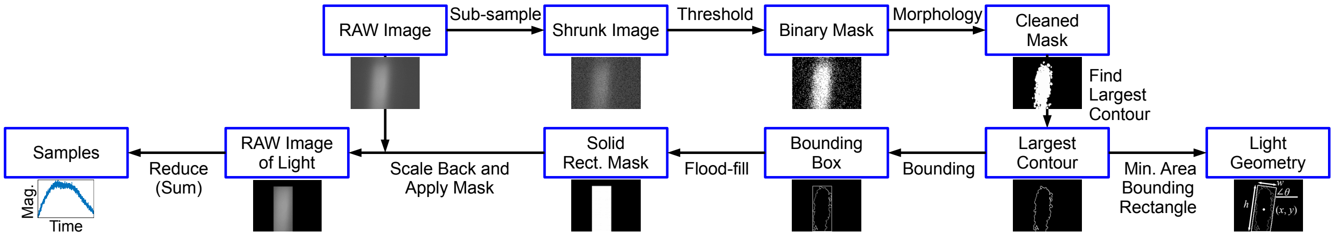


Fig. 10. Reducing spatial interference by extracting FL related pixels in the RAW camera image.

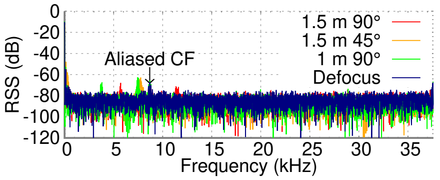


Fig. 11. Effects of defocusing.

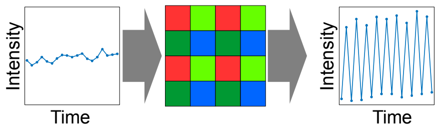


Fig. 12. Color filter array and gain mismatch.

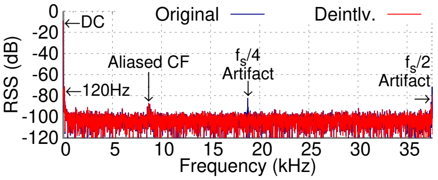


Fig. 13. Effects of de-interleaving.

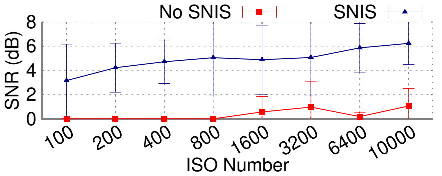


Fig. 14. Effects of ISO number, with and without SNIS.

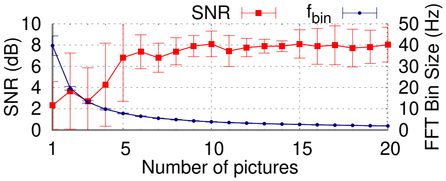


Fig. 15. SNR improvement via image concatenation.

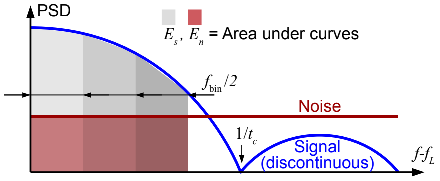


Fig. 16. SNR grows with shrinking $f_{\rm bin}$. Showing the center-aligned case for example.

inclusion of noisy pixels. It also ensures each row of the image has the same number of pixels, which provides similar summing gains. Otherwise, spurious frequency components may be introduced by gain variation across rows.

Speeding up processing by sub-sampling. Processing million-pixel images on a smartphone directly costs substantial time, which hinders real-time localization. To speed it up, we observe that the FL shape is quite regular and we can shrink the entire camera image without corrupting the contour of the FL. Therefore, we first subsample the image to 1/256 of its original size, before the thresholding and morphological processing. This allows LiTell's processing speed to grow from less than one picture per second to roughly 10 pictures per second while preserving necessary accuracy. After obtaining the bounding box, we scale up its size/position, and select the pixels inside its boundary for CF extraction.

Fig. 10 summarizes the entire workflow of SNIS. Since we use RAW images for processing, all pictures are in grayscale. A side product of SNIS is a geometrical outline of the FL, which will be used for sub-light localization (Sec. 5.3). It also isolates the light fixture containing the most pixels in the picture, thus CFs from different lights will not get mixed up. As light fixtures are a few orders of magnitude brighter than other ceiling objects, the probability of spurious light detection is extremely low. Since we find that tube FLs are the most pervasive in public buildings, the SNIS mostly targets these rectangular fixtures. For lights in other shapes, SNIS will still bound them in a rectangular box, which inevitably includes noisy background pixels and results in lower SNR.

4.2 Optimizing Camera Configurations

Isolating spatial frequency components. Many FLs come with shades or covers that bear certain spatial patterns, which creates noisy frequency components after the rolling shutter sampling. To remove the spatial artifacts, we can use a low-pass image filter. Such a filter must only be applied to spatial features alone, *i.e.*, filtering the signal spatially before it enters the camera sensor. Otherwise, it will destroy the temporal frequency components. In LiTell, we use the camera's macro mode to force the focus distance to a minimum, thus defocusing the far-away FL by acting as a spatial low-pass filter.

Fig. 11 plots an example spectrum measurement of an office FL with patterned plastic covers. The spatial features result in multiple peaks that are much stronger than the CF, and vary depending on camera viewing angle and distance. With defocusing, the impact is almost completely eliminated.

Overcoming interleaving in CMOS sensors. A camera's image sensor typically comprises millions of light sensors. The sensors of different colors are interleaved with each other [35] (Fig. 12). In practice, an FL's color may not appear white to the CMOS sensor [36]. Thus, the image sensor's odd and even rows can have quite different responses to the FL's signal, which again translates into unwanted frequency artifacts. To counteract such gain mismatch, we first obtain the mean values of even and odd rows to estimate the different gains applied to each row, and then divide the rows with the corresponding gain. Fig. 13 shows the effectiveness of de-interleaving on a Nexus 5 smartphone. We see that the interleaving induces a $f_s/4$ frequency component, which is even stronger than the characteristic frequency, but completely removed after the gain compensation.

Optimizing ISO. A camera's low-noise amplifier is controlled by ISO number. The higher the ISO, the higher the gain, and thus the more sensitive the camera is. In Fig. 14 we show how SNR of a tube FL's CF feature changes with different ISO settings. Overall, a higher ISO always results in better SNR, implying that the improved sensitivity outweighs the increased noise level. Thus, LiTell configures the camera to the highest ISO to maximize SNR. Note that this choice differs from LED-based visible light communication systems, which intentionally generate high-power flickering signals, sufficient to be captured by the camera even at low ISO [14].

4.3 Image Combining and CF Extraction

Amplifying SNR via sequential imaging combination. After the foregoing processing of a camera image, we obtain a column vector, with each element being the sum intensity of a row of pixels. To identify the peak frequency, we can simply run FFT over the vector of samples. However, we found this results in a low SNR of only around 2 dB under typical conditions.

To boost the SNR, we capture multiple camera images, and concatenate their samples into a long sequence as FFT input. Image concatenation brings two immediate benefits: *(i) Higher frequency-resolution.* Suppose f_s and N denote the sampling frequency and number of samples, respectively. Denote L as the length of samples per image, which may deviate from maximum image height due to SNIS. FFT operation's resolution equals $f_{\mathrm{bin}} = f_s/N$ [22]. To discriminate the characteristic frequency of different FLs, our empirical results show that *at least* 10 Hz resolution is needed (Sec. 5). So we need to concatenate at least $f_s/10/L$ images, considering SNIS may remove some noisy samples. For example, since Nexus 5 has a sampling rate of roughly 75 Ksps and each picture provides about 2000 samples (Table 1), we concatenate 7 pictures to get a sufficient resolution of around 5 Hz. *(ii) Higher processing gain.* It is well known that FFT can achieve a "processing gain" with more samples, *i.e.*, the peak frequency's SNR increases logarithmically (in dB) as N increases, assuming the noise power spread evenly over the spectrum (*i.e.*, white noise). Our experimental results, shown in Fig. 15, indeed verifies this quantitative relation. However, we note as N becomes large, SNR improves marginally whereas the image processing time increases substantially. We thus choose N to be the minimum value that satisfies the resolution requirement.

To see why such SNR gain exists, we build a simple analytical model. We first assume that samples are continuous. Without loss of generality, we assume the CF signal has unity power and noise has unity power spectral density (PSD). Denote the CF as f_L and number of samples as N. The PSD of signal $\mathcal{F}_s$ and noise $\mathcal{F}_n$ (in linear units) are:

$$\mathcal{F}_s(f) = \delta(f - f_L),\ \mathcal{F}_n(f) = 1 \tag{8}$$

Now consider the finite FFT bin size $f_{\mathrm{bin}} = f_s/N$. Signal and noise power collected by the bin covering f_L can be obtained by integrating the PSD across the bin:

$$E_s = 1,\ E_n = 1 \cdot f_{\mathrm{bin}} = f_s/N \tag{9}$$

$$\mathrm{SNR} = E_s/E_n = N/f_s \tag{10}$$

which indicates SNR would improve linearly with sample length (logarithmically when in dB).

However, unavoidable discontinuities exist at image boundaries. The concatenated samples thus consist rectangularly windowed image segments. Consequently, the amplitude spectrum will be convoluted by a sinc function, similar to PSK with random phase shifts. Denote number of samples per picture as n, and sampling duration of each picture as $t_c = n/f_s$, the signal PSD become:

$$\mathcal{F}'_s(f) \propto \mathrm{sinc}^2[(f - f_L)t_c] \tag{11}$$

Considering that the total power of the 2 signals should be the same, *i.e.*:

$$\int_0^\infty \mathcal{F}'_s(f)\mathrm{d}f = \int_0^\infty \mathcal{F}_s(f)\mathrm{d}f = 1 \tag{12}$$

and that

$$\int_0^\infty \mathrm{sinc}^2[(f - f_L)t_c]\mathrm{d}f = \frac{1}{t_c}\int_{-t_c f_L}^\infty \mathrm{sinc}^2(x)\mathrm{d}x \approx \frac{\pi}{t_c} \tag{13}$$

since $t_c \gg 1/f_L$ (*i.e.* $t_c f_L \gg 1$). We can then obtain an approximation of $\mathcal{F}'_s(f)$:

$$\mathcal{F}'_s(f) \approx \frac{t_c}{\pi}\mathrm{sinc}^2[(f - f_L)t_c] \tag{14}$$

To derive energy collected by the finite FFT bin, we consider 2 extreme cases: when the center of the FFT bin aligns with the signal, *e.g.* $f \in F_1 = [f_L - f_b/2, f_L + f_b/2)$, and when the edge of the FFT bin aligns with the signal, *e.g.*, $f \in F_2 = [f_L, f_L + f_b)$ or $f \in F_2 = (f_L - f_b, f_L]$. The signal power collected is:

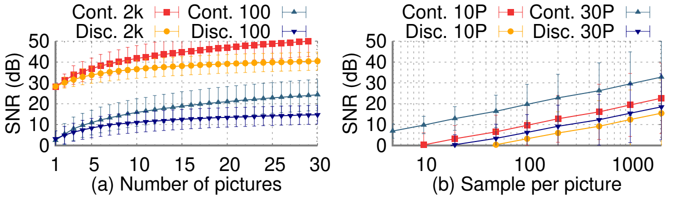
$$E'_s = \begin{cases} \dfrac{2t_c}{\pi}\displaystyle\int_{f_L}^{f_b/2+f_L} \mathrm{sinc}^2[(f - f_L)t_c]\mathrm{d}f & ,\ \text{center} \\ \dfrac{t_c}{\pi}\displaystyle\int_{f_L}^{f_b+f_L} \mathrm{sinc}^2[(f - f_L)t_c]\mathrm{d}f & ,\ \text{edge} \end{cases} \tag{15}$$

Although it is hard to see the trends of SNR in the equations, illustrations of the integrals in Fig. 16 show that: *(i)* Since the sinc function monotonically increases as f_{bin} decreases, the ratio between the area representing signal and the one for noise increases, thus SNR maximizes when $f_{\mathrm{bin}} \to 0$. *(ii)* The benefits will eventually diminish as the top of sinc function flattens, as opposed to continuous case where the SNR keeps increasing with sample length. *(iii)* When t_c decreases, the sinc function narrows but also becomes higher, thus for the same SNR f_{bin} can be larger.

We further verify the analysis with simulation, where we use a single sine tone signal and white, Gaussian additive noise. We then break the signal into segments and add random phase shifts with uniform distribution. Each simulation repeats 100 times. The results in Fig. 17(a) verify that SNR grows with number of pictures, although the discontinuous case benefits less than the continuous case. Fig. 17(b) also shows that longer sample length per picture benefits both cases. Thus, higher camera resolution is desirable.

One caveat in image concatenation is that overwhelming discontinuities occur near the edge, which again raises the noise floor on the spectrum. We identified two underlying reasons: *(i)* Camera lens vignetting, which makes outer area of the image darker than the central areas; *(ii)* smartphone not perfectly parallelizing the ceiling, resulting in one end of the image brighter than the other. We use a simple equalization procedure to mitigate these effects. Specifically, we first fit each series of samples to a 6-order polynomial function, which is common in lens correction and can capture slow varying spatial response. Then we divide the samples element-wise by this polynomial function since the distortions are essentially multiplicative scaling. This restores the samples to their undistorted form and avoids huge jumps at concatenation points.

Identifying characteristic frequency in the spectrum. After the foregoing processing on the image samples, most spurious peaks have been removed, making aliased CF the highest peak. We then run a 2-step search to identify the



(a) Number of pictures (b) Sample per picture

Fig. 17. Simulation results for SNR, against (a) number of pictures N, and (b) number of samples per picture n.

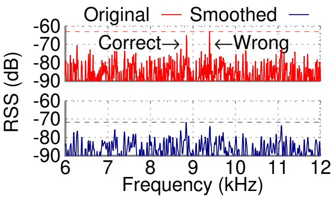


Fig. 18. Spectrum smoothing example, dashed lines show maximum.

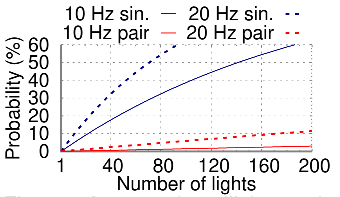


Fig. 19. Asymptotic collision probability of single and pair matching, with different tolerances.

peak frequency. We use Savitzky-Golay filter [37] on the spectrum to get a smoothed spectrum, identify a coarse peak region, and then search for the exact peak position within the corresponding region of the original spectrum. The 2-step approach helps preventing false positives when SNR is marginal. Fig. 18 shows an example, where without filtering a spurious peak can be misidentified as CF.

5 Feature Matching and Localization

LiTell uses a simple linear search algorithm to match the smartphone-extracted CF with registered CF in the database. Specifically, after obtaining an aliased copy of the CF, f_a, we first derive a collection F of possible original CFs following Eq. (7). For each $f_g \in F$, we find the registered CF with minimum difference. Finally, the f_g with minimum matching distance is considered as the FL's CF.

To guarantee accuracy, LiTell further uses history information to provide tolerance against CF feature collision. Further, we show that LiTell can also provide sub-light localization accuracy whenever an FL falls into the camera's FoV.

5.1 Multi-Light Matching

To ensure high accuracy in location matching, we leverage the fact that the user typically passes multiple lights consecutively, allowing LiTell to use a group of consecutive lights together as a location landmark as the user travels. From a high level, *multi-light matching provides an error-correction mechanism for LiTell— the strict location relation between FLs dictates that each FL only has a few candidate neighbors*. Thus, even if one FL within a group is mistaken, it may not fundamentally change the Euclidean distance (in feature space).

Asymptotic collision probability. To understand the asymptotic benefit from multi-light matching, we first analyze the probability that a group of m consecutive lights is not colliding with another group. As discussed in Sec. 2.2.1, we can reasonably assume frequency of all N_l lights follow the same normal distribution, *i.e.* $f_n \sim N(\mu, \sigma), n \in \{1, 2, \dots, N_l\}$. A Gaussian fitting shows that the 2 types of lights have $\mu = 90.25$ kHz, $\sigma = 1.62$ kHz and $\mu = 138.25$ kHz, $\sigma = 2.34$ kHz, respectively. Consider the scenario when one of 2 groups has CFs very close to the mean value μ (*i.e.*, $f_{n,1} \approx \mu$). Since in normal distribution probability density peaks at μ, this represents the worst case scenario where the collision probability is the highest. The Euclidean distance between their characteristic frequencies equals:

$$D = \sqrt{\sum_{i=1}^{m} \Delta f_{ni}^2}, D^2 = \sum_{i=1}^{m} \Delta f_{ni}^2 \tag{16}$$

(D^2/σ^2) follows $\chi^2(m)$ distribution since $\Delta f_n = f_{n,1} - f_{n,2} \sim N(0, \sigma^2)$. For LiTell to distinguish 2 groups, we need $D > a\sqrt{m}$, or $(D^2/\sigma^2) > (a\sqrt{m}/\sigma)^2$, where a is the tolerance, *i.e.* minimum distinguishable frequency difference. Denote $P(k, m)$ as the CDF of $\chi^2(m)$ distribution, then the probability of 2 groups' feature not colliding with each other equals:

$$P_g(m, a) = 1 - P\left[m\left(\frac{a}{\sigma}\right)^2, m\right] \tag{17}$$

For the case of pairs ($m = 2$), $P_g = 1 - P\left[2(a/\sigma)^2, 2\right]$. Further, the worst-case probability of one particular pair *not* colliding with *any* other pair is:

$$P_n(N_p, a) = \prod_{i=1, i\neq n}^{N_p} P_g(2, a) = P_g(2, a)^{N_p} \tag{18}$$

where N_p is number of consecutive pairs and usually $N_p < 4N_l$ since each light can at most make 8 pairs at 8 directions (including diagonals), and each pair is shared by 2 lights.

When $m = 1$, D reduces to normal random variable Δf and $P_n = \left\{\text{erfc}\left[a/(\sqrt{2}\sigma)\right]\right\}^{N_l - 1}$. The asymptotic collision probabilities for single light and pair matching under 10 Hz and 20 Hz tolerance for $\sigma = 1.62$ kHz are shown in Fig. 19.

Choosing group size. *In LiTell, to avoid excessive sampling, we set $m = 2$, i.e., using each consecutive light pair as location feature.* Yet even this small m is sufficient to achieve high accuracy. For example, with the same parameter $a = 0.01, \sigma = 1.62$ and the worst case scenario for pairs $N_p = 4N$, Eq. (18) gives $P_n \approx 98.5\%$ for a region of 100 lights, and $P_n \approx 97.0\%$ for a region of 200 lights.

Empirical validation. We verify the effectiveness of single-light and light-pair matching in an office building. The database contains each ceiling FL's coordinate, and characteristic frequency measured using the photodiode (Sec. 2). We first evaluate the single-light matching by testing over 4 regions containing 22, 28, 33 and 36 FLs, respectively (Fig. 20). The accuracy (non-collision probability) turns out to be 86.4%, 81.5%, 82.9% and 80.5%, respectively, slightly lower than the asymptotic analysis. In contrast, with light-pair matching, across 3 regions with 28, 64 and 119 lights, we achieve an accuracy of 92.9%, 91.7% and 90.8%, much higher than single-light matching. The results are summarized in Fig. 21.

We emphasize that LiTell can distinguish which light in the pair the user is currently at as long as CFs for the 2 FLs in the pair are different, thus matching pairs will not comprise localization granularity. Also note that LiTell uses a pair of characteristic frequencies as a single landmark, so wrong localization results will not propagate across different pairs.

5.2 Landmark Fusion

CF is a random feature beyond our control, and collisions will eventually happen. Fusing LiTell with landmarks that have allocated identities (*e.g.* WiFi access points and Bluetooth

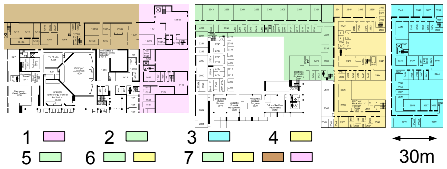


Fig. 20. Map of small-scale test areas. Area numbers are followed by regions covered.

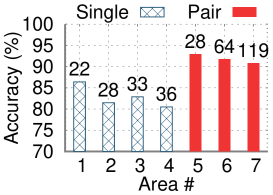


Fig. 21. Accuracy of single and pair matching. Numbers show # of lights in area.

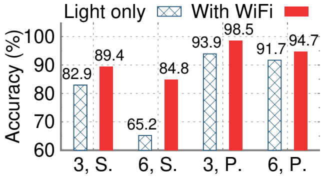


Fig. 22. Accuracy of WiFi assisted matching in area 3 (33 FLs) and 6 (64 FLs), for both single and pairs.

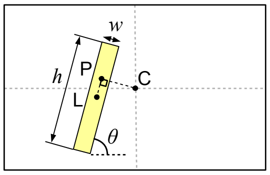


Fig. 23. Sub-light localization scheme in LiTell.

beacons) *may* make it more reliable and scalable. However, the elusive RF channel of these landmarks may nullify their benefits. Here we use WiFi to explore the accuracy improvement brought by such fusion. Although sophisticated model-based or fingerprinting-based WiFi localization schemes exist, we only adopt the simple but coarse-grained access point (AP) ID matching. We will show that this simple fusion scheme can already achieve high reliability and accuracy when combined with LiTell.

LiTell's WiFi landmark fusion works as follows. During the CF registration, LiTell also registers the BSSID with strongest RSS around each light. We only consider 2.4 GHz band due to hardware/channel heterogeneity across bands. Since WiFi's coverage is nondeterministic, in the runtime, *it is hard to define a subset of lights "near" each AP.* Rather, while LiTell is capturing the CF, it scans nearby APs and obtains a list of BSSIDs $\mathcal{B}_r$ sorted by their RSS. During FL identification, multiple candidates are selected according to frequency error of individual lights or light pairs. Then, for each candidate, LiTell looks up its corresponding BSSID in the database, gets its rank in $\mathcal{B}_r$ or discards it if not in $\mathcal{B}_r$. Finally, the candidate with the lowest rank (highest RSS) is selected as the FL's identity.

We implemented the WiFi landmark fusion on Nexus 5 and tested it in an office building, with the number of candidates empirically set to 4. An enterprise WiFi network is present the building, with one AP per every 3 to 5 lights. Fig. 22 shows that landmark fusion significantly improves accuracy of FL identification, despite the nondeterministic coverage of WiFi. Even by using individual CFs ("Ind." in the figure), LiTell's accuracy reaches nearly 90% after fusing with WiFi. For pair identification, LiTell's accuracy gets close to 100%, as it can correct occasionally erroneous matching results produced by the elusive WiFi. These results show that fusing LiTell with other landmarks with allocated identities, even ones with nondeterministic coverage, can effectively limit the collision domain for CF. Accuracy is thus decoupled from total number of lights, making LiTell more scalable.

5.3 Achieving Sub-light Granularity

Cameras are designed to produce undistorted images. From geometry relations in the image, we can derive information that enables sub-light level accuracy, which decouples LiTell's granularity from the density of FLs. To simplify the problem, we focus on the case when user holds the smartphone roughly at level position in parallel to the ceiling fixture. Then, the center of the camera image C corresponds to smartphone location, since it points to the smartphone's projection on the ceiling.

We first observe that the geometry of lighting fixtures can be easily known and added to LiTell's database. Meanwhile, the SNIS in Sec. 4.1 can provide the geometry of the light in the image. This allows us to establish the relation between physical size and number of pixels in the image as a ratio g_i. Thus, we can get the smartphone's position in terms of pixels and then map it to physical location on the 2D plane. In the case that the smartphone is held with an large pitch/roll angle, the distortion on image can be compensated as shown by [38].

Fig. 23 illustrates a smartphone's FoV which captures an FL but is not perfectly aligned with it. We use w, h and θ to denote the width, height and tilt angle of the FL's image. L denotes the center of the FL, whose physical location is known. C and P denote the center of the image and its projection onto the light's axle. The coordinate of L, C and P are denoted as (x_l, y_l), (x_c, y_c) and (x_p, y_p), respectively. By definition, $\mathbf{LP} \perp \mathbf{PC}$, thus $\mathbf{LP} \cdot \mathbf{PC} = 0$. In addition, $\tan(\theta) = (y_p - y_l)/(x_p - x_l)$, and (x_l, y_l), (x_c, y_c) are known. Consequently, we have 2 equations to solve 2 unknown variables x_p and y_p, which in turn lead to $|\mathbf{LP}|$ and $|\mathbf{PC}|$ in terms of pixels. We then translate them back to meters with g_i. The resulting physical coordinate of C gives the smartphone's position relative to the center of the light.

6 Implementation and System Efficiency

CF registration. We use the photodiode setup (Sec. 2) to collect 2 seconds of samples from each FL and identify its CF. We then manually mark the FL's location on a map, whose zero coordinate is defined at an anchoring FL near the building entrance. The map is later digitized into ⟨location, CF⟩ and entered in LiTell's database. The whole registration process takes one student volunteer around 3 hours for a medium-sized grocery store (∼1000 m^2, 162 FLs), which is an order of magnitude faster than radio-based fingerprinting [2]. Overall, *LiTell's registration process is simple, efficient and non-intrusive, and involves no management/infrastructure cost in realistic environments.*

Smartphone app. We have implemented LiTell's sampling (Sec. 3) and feature amplification mechanisms (Sec. 4) on Android. In particular, we use OpenCV [39] to implement SNIS (Sec. 4.1). Further, we prototype a simple navigation app that finds the shortest path towards a destination FL, and navigates users with basic on-screen instructions (*e.g.*, forward, left). Our experiments run on several popular Android smartphones, whose camera capabilities along with f_s (measured following Sec. 3.3) are summarized in Table 1. Unless noted otherwise, we use Nexus 5 for testing, with default ISO 10000, 7 image samples per FL, and exposure time set by the adaptive exposure scheme (Sec. 3.2).

Sever backend. Although LiTell can be implemented as a self-contained app, we separated the database as a MATLAB server for flexibility and analytics. LiTell's smartphone app

TABLE 1
Summary of camera parameters for the Android phones in our experiments.

Phone	Year	Sensor	Resolution and Framerate [a]	Estimated f_s (Sec. 3.1)	f_s (Sec. 3.3)	SNR (Sec. 7.1)
Nexus 4	2012	IMX111	3280 × 2464 (8 MP) @ 22.5 FPS	55.440 Ksps	56.3347 Ksps	4.24 dB
Moto X	2013	OV10820	4320 × 2432 (10 MP) @ 30 FPS	72.960 Ksps	60.6981 Ksps [b]	9.20 dB
Nexus 5	2013	IMX179	3280 × 2464 (8 MP) @ 30 FPS	73.920 Ksps	75.1624 Ksps	10.26 dB
Nexus 5X	2015	IMX377	4032 × 3024 (12 MP) @ 34.97 FPS	104.910 Ksps	105.1089 Ksps	6.12 dB
LG G4 (Back)	2015	IMX234	5360 × 3016 (16 MP) @ 30 FPS	90.480 Ksps	91.3079 Ksps	5.83 dB
LG G4 (Front)	2015	T4KA3	3280 × 2464 (8 MP) @ 30 FPS	73.920 Ksps	75.3481 Ksps	15.57 dB

a. Effective pixels (including inactive pixels) and maximum full-frame framerates, from sensor datasheets when available.
b. Sensor might be down-clocked. The best configuration available in its software is 1920 × 1080 @ 60 FPS, corresponding to 64.800 Ksps.

first processes the image following Sec. 3 and 4. Upon receiving the processed *samples* from the app, the server extracts aliased characteristic frequency (CF) f_a (Sec. 4.3), generates possible unaliased CFs (Sec. 3.3) and match them to a specific FL using the pair matching algorithm (Sec. 5.1). It then returns navigation instructions to the smartphone app. In case the matching confidence is low, the app will instruct the user to move to the next light for rematching.

Theoretically, it is possible to offload all the processing to the server. However, this requires sending huge RAW images (∼16 MB each) to the server, which makes wireless network the bottleneck, especially when there are multiple LiTell clients. With future deployment of faster and more efficient wireless networks and edge computing facilities [40], offloading the entire processing may make a better tradeoff for performance and energy efficiency. On the other hand, when network connection is unavailable, or when location privacy is of utmost concern, it is feasible to migrate the server side processing entirely into the app by preloading the database.

7 Experimental Evaluation

LiTell aims at combining the accuracy and robustness of VL localization with existing lighting infrastructure. In Sec. 7.1, we first evaluate LiTell's robustness against environment, devices, and human behavior. We then characterize LiTell's sub-light level localization precision in Sec. 7.2. In addition, we evaluate LiTell's energy efficiency and latency. Finally, in Sec. 7.4 we put LiTell in the wild to test its real-world performance and effectiveness.

7.1 Robustness

Light-to-camera distances. We put a smartphone under a single FL, vary the light-to-camera distance (measured using a laser ranger [41]), and measure the SNR and CF. Each measurement repeats 5 times. Fig. 24 shows that SNR decreases proportionally to distance, but remains high (> 3 dB) even at a distance of 2 m.

Fig. 24 also plots the 90th-percentile CF variation under different noise conditions, with dark green line showing ground-truth CF and gray area showing 20 Hz tolerance (Sec. 5). As long as the SNR exceeds 3 dB, the variation stays within the 20 Hz tolerance. This implies that *LiTell can robustly capture the CF features for a light-to-camera distance of up to 2 m, which is longer than ceiling-to-smartphone heights in most cases*.

It is worth noting that SNIS nearly doubles the usable range for LiTell. In some cases, we found the spectrum smoothing in Sec. 4.3 helps identify CF even under negative SNR, which indicates its importance when SNR is marginal. Due to limited SNR of smartphone cameras, we do expect LiTell to fail for ceilings much higher than our office building, especially with front-facing cameras due to their lower resolution. However, we find that a range of 2 m will already enable LiTell in many public buildings that need accurate localization service.

Temperature. Recall an FL's electronic component ratings and hence stability of its CF is affected by temperature. We evaluate the impact in a small office, where we control the temperature settings via a thermostat and verify the change using a `DS18B20` sensor [42] (0.1°C precision). Fig. 25 shows that the CF decreases as temperature increases, which is consistent with the heat-up behavior in Fig. 6. However, within 4°C temperature gap, the CF fluctuates by less than 20 Hz, well within LiTell's stability tolerance (Sec. 5). Since the temperature in most public buildings tends to be tightly regulated, *LiTell is robust across temperature changes in typical regulated indoor environments*.

Meanwhile, we also tested LiTell in an outdoor parking ramp and do find the CF becomes unusable due to large (> 10°C) air temperature variations. Currently, we see offices and shopping centers as LiTell's target application scenarios, where room temperature is always tightly regulated. For example, during a period covering winter and early summer, CFs of FLs in our office building did not drift significantly enough for re-registration to be necessary. However, for places with unregulated temperature, it might be possible to derive a CF-vs-temperature model for each FL model, and use it to compensate the CF database.

Ambient light. To evaluate LiTell's robustness against ambient light interference, we put the smartphone 1.5 m away from a tube FL close to a window. The experiment started on a sunny day at 2:30 pm until sunset. A Lux meter [43] is placed nearby to track ambient light intensity. Fig. 26 shows that LiTell consistently captures the CF with more than 3 dB SNR except for extreme cases. In contrast, when LiTell's SNIS is disabled, the SNR drops to 0 dB under sunlight interference, making it impossible to discriminate CF from noise. Therefore, *SNIS not only improves SNR, but also makes LiTell more robust under ambient sunlight interference.* Meanwhile, the SNR is negatively impacted by the sunlight. when coupled with other factors (*e.g.* higher ceilings), LiTell might fail for lights installed near windows on a sunny day. However, we expect this to be a rare case, and does not prevent operation in internal regions of the building.

Different buildings. To test LiTell's robustness and accuracy in the wild, we set up LiTell in 3 different venues (Fig. 27): an office building (∼9000 m^2, 119 FLs, ceiling height ∼3.0 m, ∼5 m FL separation), an underground parking

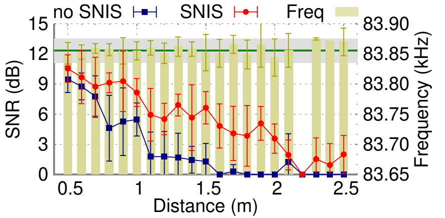


Fig. 24. SNR (lines/points) and CF (yellow bars) detection accuracy.

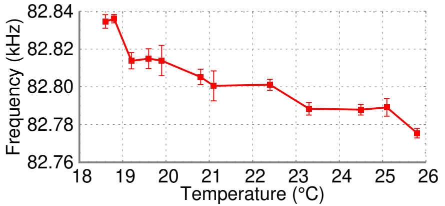


Fig. 25. Impact of room temperature.

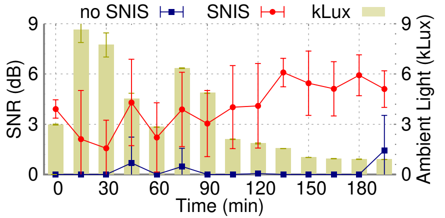


Fig. 26. SNR variations with ambient light intensity.

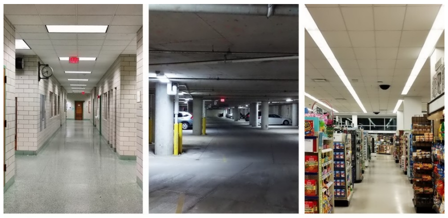


Fig. 27. Photos of buildings. From left to right: office building, parking lot, grocery store.

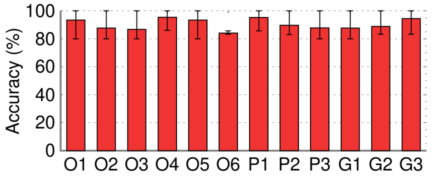


Fig. 28. Accuracy across buildings. "O": office building, "P": parking lot, "G": grocery store.

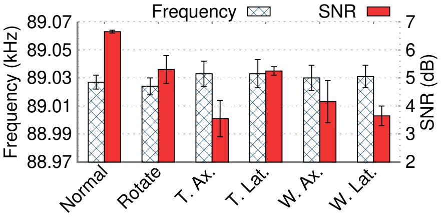


Fig. 29. Impact of human behavior.

lot (∼2800 m^2, 91 FLs, ceiling height ∼2.8 m, ∼8 m FL separation), and a medium-sized grocery store (∼1000 m^2, 162 FLs, ceiling height ∼4.0 m, contiguously placed FLs). The experiments were conducted approximately two months after FL registration, with landmark fusion disabled to explore the worst case performance.

Fig. 28 plots the FL identification accuracy, with error bars showing min/max across 10 trials. Across all venues, LiTell can discriminate the FLs with mean accuracy of 90.3% and a small variation of 11% from the best to the worst, which is very close to results obtained in controlled environments (Fig. 21), despite different light models, distribution and ceiling heights. In particular, FLs in the grocery store are closely placed in lines 2 m away from the phone, resulting in multiple lights being captured simultaneously. However, *LiTell's SNIS mechanism can successfully isolate the lights, and only include the nearest one with the largest area in the image.* Additionally, *busy customers in the grocery store and moving cars in the parking lot did not degrade LiTell's accuracy*, since the CF feature is deterministic. The results also show *temperature in most indoor environments is stable enough for LiTell to operate.*

Smartphone models. LiTell's CF extraction performance may be affected by different smartphone models, with varying resolution and speed (which determine the rolling-shutter sampling rate) and sensor quality (which determines SNR). Whereas it is infeasible to exhaust all smartphones, we test 5 of them representing different generations that were launched in the past 5 years, by sampling a tube FL 1 m away. Each phone is equipped with the latest stock ROM available at the time. Table 1 summarizes the measurement results. Despite the different camera capabilities, all the phones can detect the CF with high SNR ($>$ 6 dB except for Nexus 4 and back camera of LG G4). It is worth noting that Nexus 4 and Moto X do not allow fine-grained exposure time or ISO configuration, and only allow JPEG imaging which may distort the CF features due to non-linear processing [44]. The SNR of detected CF is relatively lower, but still exceeds LiTell's 3 dB minimum requirement, based on LiTell's other camera optimization mechanisms. The SNR also seems to be affected by multiple factors, instead of just monotonically growing with number of pixels. These factors might include sensor model, lens configuration and f_s (since frequency response changes with f_s, Sec. 3.2). We also notice that the front camera of LG G4 works with sufficient margin in SNR. We expect in the future more and more smartphones will have high-quality selfie cameras that can support LiTell.

Human behavior. We also verify that LiTell can work consistently across different users, whose usage behaviors may vary. In the experiment, we sample an FL from various directions and while walking. To cover extreme scenarios, we hold the smartphone within 1 m to the FL, so the viewing angle can change rapidly with position. Then we create the following test cases: *(i) normal test case:* the phone is placed level right under the FL and aligned with its tube direction. *(ii) rotate case:* the phone is misaligned with the tube by about 30°; *(iii) tilt axial / lateral case* ("T. Ax." and "T. Lat."): the phone photos the fixture sideways from the axial and lateral direction of the tube; *(iv) walking axial / lateral case* ("W. Ax." and "W. Lat."): the user walks across the FL along its axial and lateral directions.

Fig. 29 shows that LiTell successfully detects the CF in all the cases, with 90th-percentile deviations of less than 10 Hz. The SNR is consistently above 3 dB by a large margin. This shows that *LiTell's CF detection and amplification mechanisms make it robust to the extra variation in light intensity and effects caused by elusive usage behaviors.* More specifically, the sampling and processing work fast enough to ensure the CF remains visible during capture of 7 images. LiTell's SNIS is also robust against placement and shape, and the macro defocusing mechanism can successfully filter spatial features, leaving only the temporal frequencies in the spectrum.

7.2 Localization Precision

To evaluate the granularity of sub-light localization, we use a light fixture with a 1.2-meter-long tube FL, 1.8 m above the phone. We place the phone on a motored slider [45] and move it to 20 locations 5 cm apart from each other in a 1.1 m × 0.4 m area (limited by slider length and camera FoV). We record the localization error of 10 attempts, and then repeat the experiment with a user holding the phone while "walking" on the spot. Fig. 30 shows that LiTell can achieve 10 cm accuracy 90% of the time if placed still and level. When held by a walking user, the phone is no longer strictly level, which affects LiTell's geometrical model (Sec. 5.3). But the median precision is still as high as 15 cm and 90% at 25 cm.

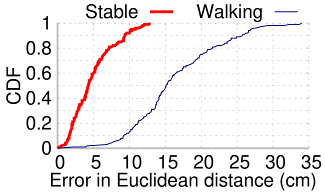


Fig. 30. Performance of sub-light localization.

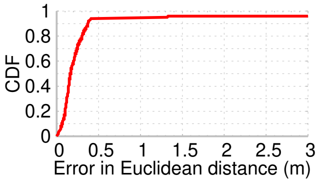


Fig. 31. Combined performance of FL identification and sub-light localization.

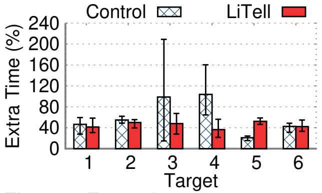


Fig. 32. Extra time spent on navigation by users. Error bars show min/max values.

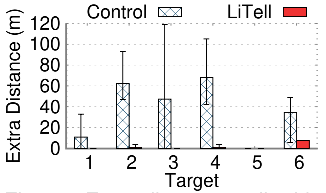


Fig. 33. Extra distance walked by users. Error bars show min/max values.

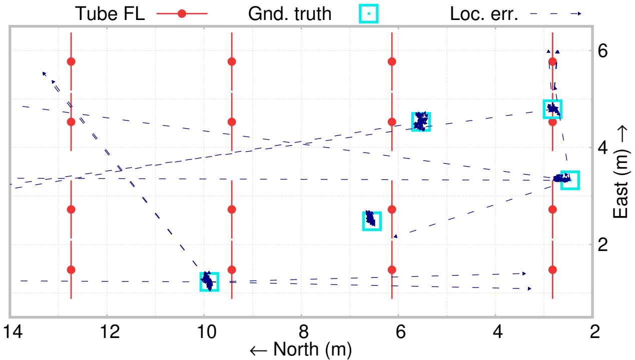


Fig. 34. Sub-light localization results.

As the distance gets larger, the physical distance that a pixel covers also increases linearly, so the error will grow linearly with distance. Considering most buildings have heights less than 5 m and users usually hold the smartphone at least 1 m above the floor, the error of sub-light localization should remain below 0.5 m in most cases.

To profile the combined performance of FL identification and sub-light localization, we selected a rectangular open-office space with 23.7 m × 6.4 m area and 3.1 m ceiling height, in which the ground-truth location can be conveniently measured. We registered 56 unique 1.2-meter-long tube FLs (2 per fixture) in the area, which are arranged in rows 3.3 m apart. We then picked 5 random spots that have the light in the smartphone camera's FoV when laid flat. For each spot, we hold the smartphone 1.7 m below the light, against a piece of fixed furniture and record 50 localization results.

Fig. 34 zooms in on the localization results, with errors shown in arrows. Except for occasional misidentifications, all the localization results are tightly-packed around the ground truth, thanks to deterministic visible light channel high-resolution of the camera. Fig. 31 further shows the end-to-end localization precision, with median at 17 cm and 90th-percentile at 37 cm. The long tail is caused by misidentified FLs, which are located far away from the ground-truth.

7.3 Latency and Efficiency

On Nexus 5, our LiTell implementation takes 92 ± 35 ms to process each image, and around 700 ms for 7 images. We find that a majority of the processing time is due to suboptimal implementation (Java instead of native C) of the gluing operations, which cost around 70 ms per image. We hence believe that with proper optimization, the computation can be done within one frame interval (33 ms for 30 FPS), allowing 7 pictures to be captured and processed within 233 ms. For the database server, all computation (including CF extraction and matching, Sec. 3.3, 4.3 and 5.1) takes 112 ms per request on average while running on a laptop with Intel Core i5 3340M CPU. Experiments in our office building show an average end-to-end location query-response time of 1.62 s, which is tolerable for most applications.

We profile the energy consumption of the LiTell app using the Monsoon Power Monitor [46]. The Nexus 5 consumes 741.84 ± 11.52 mW when idle, 2001.60 ± 28.50 mW when the camera's viewfinder is active, and 2669.69 ± 29.47 mW when the LiTell app is capturing, processing and sending simultaneously. As Nexus 5 has a 8.74 Wh battery [47], and a localization attempt only lasts ∼2 s and consumes ∼1.20 mWh energy, the smartphone can support thousands of localization attempts per battery cycle.

We build LiTell on top of cameras due to their availability on COTS smartphones. This leads to unnecessarily high power consumption as information from most pixels is discarded and wasted. In the future, however, efficient and high-speed photodiodes used by visible light communications (VLC) may become increasingly available on COTS devices. LiTell can be adapted to leverage photodiodes as high SNR, alias-free and energy-efficient light sensors. On the other hand, LiTell could use the slow but low-power ambient light sensors to determine when to turn the camera on, which may help reduce energy consumption considerably.

7.4 User Study and Field Tests

To test LiTell's effectiveness in indoor navigation, we recruited 10 volunteer users who frequent the venues, but may not have precise knowledge of all the points of interest (POI, *e.g.*, room number). Average height of the users is around 1.7 m. We randomly select 6 targets in the office building in Fig. 27: 1, 4 and 5 are on the 1st floor, while 2, 3 and 6 are on the 2nd. In addition, 5 and 6 are in a corner of the building not far from the entrance. The oracle routes vary from 35 m to 132 m including the stairs, each containing at least one turn. The trials are conducted at different times of the day, with human activities around most of the time. In each test, we ask a participant to start from the building entrance, and find a random POI which may be on a different floor. We perform 3 sets of trials: *(i) Controlled test:* A user finds the POI with common sense or help available (maps on walls, direction signs and people walking by). *(ii) LiTell test:* A user (different from the controlled one) uses the LiTell navigation app to find the same POI. *(iii) Oracle baseline*. After the above two tests, we let all users know the shortest path and walk directly to the POI.

In each test, we follow the user and record the walking time and path on a map. Then, we derive the *extra distance and percentage of extra time spent on navigation for test (i) and (ii), by subtracting the oracle baseline from them.* We use percentage in walking time to compensate for different walking speeds. Since the prototype LiTell implementation uses back cameras, the users sample the light with the phone facing down when requested by the app, and then turn the phone upright again to read instructions, which causes some unnecessary overhead in time consumption.

Fig. 32 shows the extra time cost *w.r.t.* the oracle. Compared with the controlled tests, LiTell saves different amount of time depending on how hard the target can be found. Particularly, target 5 is only about 35 m away from the entrance point, and even the controlled group can easily find it with only 7 seconds of extra time compared with the oracle baseline. However, when the user is unfamiliar with the target POI, LiTell can save over 50% time on average compared with the control group, for both single-floor and cross-floor navigation. On the other hand, LiTell's extra time cost *w.r.t.* oracle is consistent at roughly 40% across all targets, regardless of distance, usage habits, area, number of turns or floors. This again confirms that *LiTell's CF extraction mechanism is robust against practical usage scenarios*.

Fig. 33 further plots the extra distance cost *w.r.t.* the oracle. Extra distance needed by LiTell is typically only a few meters, owing to its high localization precision. Interestingly, we find even for the users who frequent the building, their trajectory can be highly suboptimal, and LiTell can help prevent them from traveling to wrong places or along detour paths. The average distance saving can be 50–70 m for POI 2, 3 and 4, and up to 120 m for certain users. Compared with such distance saving, the time saving in Fig. 32 might seem relatively small. This is mainly due to LiTell's processing overhead in each localization attempt. We believe an optimized implementation (Sec. 6) can cut the time overhead and enable real-time navigation.

8 Limitations and Future Work

Ubiquity of FL vs. LED. While LEDs are likely to replace FLs eventually, current progress is slow. The main reason is that FL has already saturated the market, and its low cost and availability still lead to new deployment. Moreover, LED hardware still bears certain limitations, such as lower color quality, narrow beam angle and lower lumen output [48]. Thus, the US Department of Energy [16] predicts that FLs will continue dominating the market, occupying more than 60% of the market till 2020, and 30% to 40% till 2030. From our experience, buildings with any LED lighting remains scarce for now, and even in these buildings the major force of lighting is still FL. In addition, LEDs used for commercial lighting differ from "smart" bulbs that can send unique beacons. Due to cost, size and management issues, we expect "smart" bulbs and regular LED bulbs or even FLs to be deployed together in the future, and systems that work with regular LED/FLs will still have unique values.

Most high-power LEDs are driven by constant current drivers, which oscillate much like FL drivers [49]. Since LEDs are driven by DC, manufacturers tend to add output capacitors to suppress the switching noise of the driver, which attenuate the characteristics frequency (CF) severely. From our measurements, some LEDs exhibit CFs that are 60 dB weaker than the 120 Hz and DC signal, while others do show very strong CF. Due to limited SNR of cameras, current LiTell system may not work with all varieties of LEDs. However, once high-SNR, high-speed light sensors (*e.g.* photodiodes used in VLC [50]) are available on smartphones, LiTell will also work with majority of the LEDs.

Long-term CF drifts. Across our nearly half-year observation, the CFs of experimented FLs did not change significantly. But over a longer term, the CF change might eventually happen, *e.g.* due to severe aging or replacement of worn-out lights. When such change occurs, LiTell will either give low confidence in matching, or find user's trajectory to be discontinuous. Users may choose to allow the localization server to analyze such matching confidence, and then isolate and correct for the changes.

Seamless operation. Nearly all VL localization systems need line-of-sight channel to work. Together with limited FoV of cameras (usually $\sim 70^\circ$ [51]), this makes it hard to achieve continuous and seamless coverage (*e.g.* when device is in the pocket or light falls out of FoV), and may preclude applications that require silent or continuous localization. On the other hand, the high accuracy and robustness of VL localization make them ideal for robotics or applications that are actively initiated by the user, such as navigation and inventory search. As a result, LiTell and other VL localization systems are highly valuable and can complement WiFi-based localization systems.

Co-located lights. When multiple lights are installed closely to each other, especially in parallel, LiTell's SNIS (Sec. 4.1) may have difficulty telling them apart. Consequently, multiple frequencies will be extracted from the spectrum. However, LiTell will only pick the strongest one, which still maps to the correct light. Potentially, such co-located lights can also enrich the CF feature, and allow multi-light matching (Sec. 5.1) without traveling between lights. Meanwhile, sub-light level accuracy might suffer as the geometry of the light is wrong. This can be avoided if the geometry of the whole fixture, instead of the individual lights, is registered in the database.

9 Conclusion

In this paper, we explore the feasibility of using unmodified FL fixtures to build a robust indoor localization system. We design and implement LiTell, a system that can discriminate subtle differences in the weak, high-frequency characteristics of FLs' emission, utilizing COTS smartphones' cameras augmented with customized sampling/amplification algorithms. Our field tests show that LiTell holds the promise as a ready-to-use, easy-to-deploy indoor localization system that is robust against environment and user habits. We believe LiTell leads to a new direction for robust, infrastructure-free indoor localization with visible light.

Acknowledgments

The authors would like to thank Walgreens for providing the test field. The work reported in this paper was supported in part by the NSF under Grant CNS-1506657, CNS-1518728, CNS-1343363 and CNS-1350039.

References

[1] ABI Research, "Indoor Location in Retail: Where Is the Money?" 2013.

[2] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned," in *Proc. of ACM/IEEE IPSN*, 2015.

[3] D. Lymberopoulos and J. Liu, "The Microsoft Indoor Localization Competition: Experiences and Lessons Learned," *IEEE Signal Processing Magazine*, vol. 34, no. 5, 2017.

[4] P. Bahl and V. Padmanabhan, "RADAR: an In-Building RF-Based User Location and Tracking System," in *Proc. of IEEE INFOCOM*, 2000.

[5] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based Indoor Localization," in *Proc. of ACM MobiSys*, 2012.

[6] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, 2015.

[7] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor Localization Without the Pain," in *Proc. of ACM MobiCom*, 2010.

[8] J. Xiong and K. Jamieson, "ArrayTrack: A Fine-grained Indoor Location System," in *Proc. of USENIX NSDI*, 2013.

[9] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," in *Proc. of ACM SIGCOMM*, 2015.

[10] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the Limit of WiFi Based Localization for Smartphones," in *Proc. of ACM MobiCom*, 2012.

[11] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng, "I Am the Antenna: Accurate Outdoor AP Location Using Smartphones," in *Proc. of ACM MobiCom*, 2011.

[12] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A Visible Light Based Positioning System," in *Proc. of USENIX NSDI*, 2014.

[13] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Wearables Can Afford: Light-weight Indoor Positioning with Visible Light," in *Proc. of ACM MobiSys*, 2015.

[14] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proc. of ACM MobiCom*, 2014.

[15] J. Armstrong, Y. Sekercioglu, and A. Neild, "Visible Light Positioning: a Roadmap for International Standardization," *IEEE Communications Magazine*, vol. 51, no. 12, 2013.

[16] U.S. Department of Energy, "Energy Savings Forecast of Solid-State Lighting in General Illumination Applications," Aug. 2014.

[17] Q. Xu, R. Zheng, and S. Hranilovic, "IDyLL: Indoor Localization Using Inertial and Light Sensors on Smartphones," in *Proc. of ACM UbiComp*, 2015.

[18] D. Parikh and G. Jancke, "Localization and Segmentation of A 2D High Capacity Color Barcode," in *Proc. of IEEE WACV*, 2008.

[19] S. Gupta, M. S. Reynolds, and S. N. Patel, "ElectriSense: Single-point Sensing Using EMI for Electrical Event Detection and Classification in the Home," in *Proc. of ACM UbiComp*, 2010.

[20] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. Reynolds, and S. Patel, "Disaggregated End-Use Energy Sensing for the Smart Grid," *IEEE Pervasive Computing*, vol. 10, no. 1, 2011.

[21] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin, "RollingLight: Enabling Line-of-Sight Light-to-Camera Communications," in *Proc. of ACM MobiSys*, 2015.

[22] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing, 3^{rd} Edition*. Prentice-Hall, 2009.

[23] C. Zhang and X. Zhang, "LiTell: Robust Indoor Localization Using Unmodified Light Fixtures," in *Proc. of ACM MobiCom*, 2016.

[24] R. I. Sasaki, "The impact of electronic ballast compact fluorescent lighting on power distribution systems," *Purdue ECE Technical Reports*, 1994.

[25] Philips Electronics N.V., "Self Oscillating 25W CFL Lamp Circuit," 2000.

[26] R. Spence and R. S. Soin, *Tolerance design of electronic circuits*. World Scientific, 1988.

[27] R. Narasimhan, M. D. Audeh, and J. M. Kahn, "Effect of electronic-ballast fluorescent lighting on wireless infrared links," *IEE Proceedings - Optoelectronics*, vol. 143, no. 6, 1996.

[28] S. M. Kashmiri, K. A. Makinwa *et al.*, *Electrothermal frequency references in standard CMOS*. Springer, 2013.

[29] "PicoScope," https://www.picotech.com/products/oscilloscope.

[30] C. K. Liang, L. W. Chang, and H. H. Chen, "Analysis and Compensation of Rolling Shutter Effect," *IEEE Transactions on Image Processing*, vol. 17, no. 8, 2008.

[31] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, 1986.

[32] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979.

[33] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, 1987.

[34] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[35] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, 2005.

[36] E. Y. Lam, "Combining gray world and retinex theory for automatic white balance in digital photography," in *Proc. of International Symposium on Consumer Electronics (ISCE)*, 2005.

[37] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical Chemistry*, vol. 36, no. 8, 1964.

[38] S. Zhu and X. Zhang, "Enabling High-Precision Visible Light Localization in Today's Buildings," in *Proc. of ACM MobiSys*, 2017.

[39] "OpenCV4Android," https://opencv.org/platforms/android/.

[40] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, 2016.

[41] "Bosch DLE 40 Professional," http://www.bosch-pt.com.ph/ph/en/laser-measure-dle-40-131500-0601016370.html.

[42] C. Zhang, "Access Dallas 1-wire bus on your PC with simple and cheap hardware," https://github.com/dword1511/onewire-over-uart.

[43] "Extech 407026," http://www.extech.com/display/?id=14211.

[44] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, 1992.

[45] "Axis360 Pro," http://cinetics.com/axis360-pro/.

[46] "Monsoon Power Monitor," https://www.msoon.com/LabEquipment/PowerMonitor/.

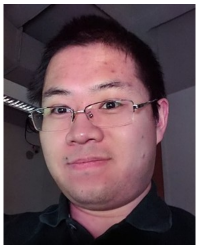
[47] "Nexus 5 Teardown," https://www.ifixit.com/Teardown/Nexus+5+Teardown/19016.

[48] U.S. Department of Energy, "Cost-Effectiveness of Linear (T8) LED Lamps," May. 2014.

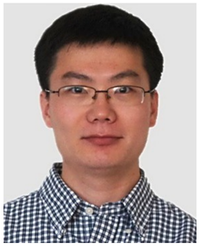
[49] H. van der Broeck, G. Sauerlander, and M. Wendt, "Power driver topologies and control schemes for LEDs," in *Proc. of IEEE APEC*, 2007.

[50] J. Zhang, X. Zhang, and G. Wu, "Dancing with Light: Predictive In-frame Rate Selection for Visible Light Networks," in *Proc. of IEEE INFOCOM*, 2015.

[51] B. Klug, "Understanding Camera Optics & Smartphone Camera Trends," https://www.anandtech.com/show/6777.



Chi Zhang received his B.E. degree from the School of Optical and Electronic Information, Huazhong University of Science and Technology, China in July 2013, and Ph.D. degree from the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, USA in December 2017. He is currently a postdoc scholar at University of California San Diego. His research interests include low-power & ubiquitous sensing, Internet-of-things, low-power wireless networking, and scalable wireless power delivery.



Xinyu Zhang received his B.E. degree from the Harbin Institute of Technology, China in 2005, M.S. degree from the University of Toronto, Canada in 2007, and Ph.D. degree from the University of Michigan, USA in 2012. From 2012 to 2017, he was an Assistant Professor in the Department of Electrical and Computer Engineering, University of Wisconsin-Madison. Since Aug. 2017, he has been an Associated Professor in the Department of Electrical and Computer Engineering, University of California San Diego. His research focuses on the systems and experimental aspects of 5G millimeter-wave networks, and wireless sensing technologies for IoT applications.